GETHOST

Make deep copies of static return data before issuing a new API call

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-23

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5585 bytes

Attack Category	• Identity Spoofing • Malicious Input
Vulnerability Category	• Threading and synchronization problem • Buffer Overflow
Software Context	• Networking
Location	• netdb.h
Description	The gethostbyname() function returns a structure of type hostent for the given host name. Here name is either a host name, or an IPv4 address in standard dot notation, or an IPv6 address in colon (and possibly dot) notation. (See RFC 1884 for the description of IPv6 addresses.) If name is an IPv4 or IPv6 address, no lookup is performed and gethostbyname() simply copies name into the h_name field and its struct in_addr equivalent into the h_addr_list[0] field of the returned hostent structure. If name doesn't end in a dot and the environment variable HOSTALIASES is set, the alias file pointed to by HOSTALIASES will first be searched for name (see hostname(7) for the file format). The current domain and its parents are searched unless name ends in a dot. The gethostbyaddr() function returns a structure of type hostent for the given host address addr of length len and address type type. The only valid address type is currently AF_INET. Note: The functions gethostbyname() and gethostbyaddr() may return pointers to static data, which may be overwritten by later calls. Copying the struct hostent does not suffice, since it contains pointers; a deep copy is required. This is the real problem here, as exposed in multithreaded applications.
APIs	**Function Name** / **Comments** gethostbyaddr

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	gethostbyname
Method of Attack	Attacker can return forged results to allow spoofing attack or arbitrarily large results to allow buffer overflow attack.
Exception Criteria	None known.
Solutions	**Solution Applicability**: Generally applicable **Solution Description**: Ensure that if there are multiple calls to the APIs listed that deep copies are made of the static return data, before reissuing a new API call. **Solution Efficacy**: Generally effective.
Signature Details	struct hostent *gethostbyname(const char *name); struct hostent *gethostbyaddr(const char *addr, int len, int type);
Examples of Incorrect Code	

Examples of Corrected Code	`// Just use the thread safe gethostbyname_r function`
	`res=gethostbyname_r(hostname,`
	`(struct hostent *)buf,`
	`(char *)buf + sizeof(struct hostent),`
	`CURL_HOSTENT_SIZE - sizeof(struct hostent),`
	`&h, /* DIFFERENCE */`
	`&h_errnop);`

Source References	• Rough Auditing Tool for Security (RATS)[2] • U.S. Dept. of Energy Computer Incident Advisory Capability. H-13: IBM AIX(r) Security Vulnerabilities (gethostbyname,lquerypv)[3] (1996). • Sun Microsystems. Programming Guidelines.[4] • Solaris Man Pages[5]. gethostbyname(3NSL) (2002).
Recommended Resource	
Discriminant Set	**Operating System**: • Windows **Languages**: • C • C++

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com